future internet

MDPI

Article

A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing

Adrien Bonguet † and Martine Bellaiche *

Computer Engineering and Engineering Software, École Polytechnique de Montréal, QC H3T 1J4, Canada; adrien.bonguet@polymtl.ca

* Correspondence: martine.bellaiche@polymtl.ca; Tel.: +1-514-340-4711 (ext. 4679)

† Current address: 2900, boul. Édouard-Montpetit Montréal, QC H3T 1J4, Canada.

Received: 19 July 2017; Accepted: 1 August 2017; Published: 5 August 2017

Abstract: Cloud Computing is a computing model that allows ubiquitous, convenient and on-demand access to a shared pool of highly configurable resources (e.g., networks, servers, storage, applications and services). Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are serious threats to the Cloud services' availability due to numerous new vulnerabilities introduced by the nature of the Cloud, such as multi-tenancy and resource sharing. In this paper, new types of DoS and DDoS attacks in Cloud Computing are explored, especially the XML-DoS and HTTP-DoS attacks, and some possible detection and mitigation techniques are examined. This survey also provides an overview of the existing defense solutions and investigates the experiments and metrics that are usually designed and used to evaluate their performance, which is helpful for the future research in the domain.

Keywords: cloud computing; denial-of-service; security; countermeasures

1. Introduction

Cloud Computing includes both what is delivered as a service over the internet and the hardware behind those services. Resources can be provisioned and released very easily, requiring little if any intervention from the provider. From the user's point of view, Cloud infrastructures seem to provide infinite resources, which can be adapted to one's needs. For example, a small start-up company may lack the need or the financial resources required to buy many computing resources, but may want to leave its options open for a future expansion, if successful, making Cloud Computing particularly appropriate in such a case. In this context, the company would simply pay for what is actually used given that resources can be released when they are no longer required.

In a Cloud network, users do not own the computing servers. They can access numerous services without the burden of Cloud management and their data can be accessed by way of many devices (such as smart phones, sensors, tablets, etc.).

More generally, the main features of Cloud Computing are the following [1]:

1. *Large-Scale*: to satisfy the customers' demands, companies like Amazon, IBM, Microsoft, Yahoo and Google own hundreds of thousands of distributed servers.
2. *Resource pooling*: providers serve multiple customers with provisional and scalable services. These services can be transparently adjusted to the clients' needs.
3. *Ubiquitous network access*: users can access services anywhere, through any kind of terminal.
4. *Rapid elasticity*: users can increase and release their requests quickly and dynamically.
5. *On-demand self-service*: since a Cloud infrastructure is a large pool of resources, users can buy according to their needs. The provider automatically supplies services and associates resources to the Cloud user, as requested.

6. *High extensibility*: the scale of a Cloud infrastructure can be extended to meet the increasing requirements of customers.

Security in Cloud Computing is critical when developing services. Updating the operating systems of virtual machines, ensuring availability, isolating users' individual data, implementing authentication mechanisms, encryption or configuring VPN and VLAN are but a few examples of what needs to be considered [2]. Here is a list of the security aspects that challenge Cloud Computing.

1. *Identity, Authentication, Authorization*
   Identity enables characterizing a user through the use of a login. Authentication is used to verify the user's credentials. This is done in a secure, trustworthy and manageable manner [3]. When authentication is complete, the Cloud authorization verifies the user's rights. Guidance includes a centralized directory, identity management, privileged user and access management, role-based access control and separation of duties among main features. In addition, the service provider can frequently offer a free trial period. For example, in the summer of 2012, attackers (users for a free period) accessed Mat Hona's data (writer for Wired Magazine) Apple, Gmail and Twitter accounts [3]. They erased all his personal data in those accounts.
2. *Confidentiality*

   A malicious attacker in a virtual machine can listen to another virtual machine [4]. An attacker can very easily identify the data center of the Virtual Machine (VM) and can also obtain information about the IP address and the domain name of the data center. In addition, a VM can extract private cryptographic keys being used in other VMs on the same physical server, which subsequently implies the risk of data leakage [3]. It is thus important to protect the confidentiality of VM data. For example, the Amazon EC2 platform (Seattle, Washington, WA, USA) [5] was vulnerable to confidentiality issues [4]. However, now, with Amazon Web Service (AWS), the client has the option to manage their own encryption keys [6].
3. *Integrity*

   Phishing, fraud and exploitation of software vulnerabilities, traffic hijacking can eavesdrop activities and transactions, manipulate data, return falsified information and redirect clients to illegitimate sites. Similarly, weak interfaces and Application Program Interface APIs cannot protect users from accidental or malicious attempts [3]. For example, Hewlett-Packard (Palo Alto, California, CA, USA) proposes an Integrity Virtual Machines Architecture [7].
4. *Isolation*

   Cloud Computing must have a level of isolation among all the VM data and the hypervisor [8,9]. In Infrastructure as a Service (IaaS), it means isolating VMs' storage, processing memory and access path networks. In Platform as a Service (PaaS): running services and API calls must be isolated. Moreover, in Software as a Service (SaaS): isolation amongst transactions must be achieved.
5. *Availability*

   Illegitimate users consume much of the victim's processing power, memory, disk space or network bandwidth. It also causes system slowdowns, which prevents legitimate users from using the service. Consequently, the VM becomes unavailable, causing a Denial of Service (DoS) or Distributed Denial of Service (DDoS). For example, a DDoS attack with compromised Internet of Things devices happen on Dyn (DNS infrastructure) [10] and paralysed some cloud computing-based sites such as GitHub and Airbnb [11].

This survey focuses on DoS and DDoS attacks and defenses applied to Cloud Computing availability. It will demonstrate that DoS and DDoS attacks (specifically XML-DoS and HTTP-DoS) present a serious threat to Cloud Computing, with many vulnerabilities, originating from various types of attacks and attackers, the latter originating from various types of attacks and attackers. This paper presents the design experiment and the metrics used to evaluate DoS and DDoS defenses.

Our contributions to this survey are to present the following:

- DoS and DDoS attacks targeting Cloud availability;
- a description of the specific XML-DoS and HTTP-DoS attacks;
- the defenses applied to DoS and DDoS attacks in Cloud Computing;
- the specific defenses against XML-DoS and HTTP-DoS attacks;
- a summary of how to evaluate such defenses.

So far, the features of Cloud Computing and certain security aspects were presented. The remainder of this paper is organized as follows: Section 2 describes the work related to the classification of security issues, Section 3 identifies the possible attacks and attackers, Section 4 focuses more specifically on DoS and DDoS applied to Cloud Computing and Section 5 examines possible detections and mitigations of DoS and DDoS attacks. Furthermore, Section 6 identifies the experimentation and the metrics to evaluate the defenses and Section 7 offers conclusions.

2. Related Work

The new paradigm introduced by Cloud Computing creates new security challenges. Therefore, a number of scientific contributions were made in this field during the past few years. Much work has been done to identify threats and vulnerabilities and new frameworks and strategies were created to address such problems. Furthermore, these security concerns are likely to increase in the coming years due to the progressive migration of companies and individuals to Cloud infrastructures. The following is a review of some of the Cloud security surveys that were recently published.

Grobauer et al. [12] exposes vulnerabilities associated with Cloud Computing. For example, the vulnerabilities are (1) VM escape; (2) session riding and hijacking; (3) insecure or obsolete cryptography; (4) unauthorized access to management interface; (5) Internet protocol vulnerabilities and (6) data recovery vulnerability. The authors specify that the current security metrics are not adapted to Cloud infrastructures, so that new metrics standards must be developed for greater security. Although they clarify indicators of Cloud-specific vulnerabilities, no solutions are presented to solve them.

Gonzalez et al. [9] identify, classify, organize and quantify the security taxonomy-architecture: network configuration, hosts and virtualization issues, applications and services, data security and storage, security management as well as identities and access to Cloud Computing. In addition, the authors present security concerns and solutions using pie charts in order to show the representativeness of each group with identified references. They identify that the security problems associated with virtualization are the most seriously evaluated at 12%, but the research on solutions for this aspect is only 3%. They propose developing new mechanisms to isolate VMs, since proper isolation between VM must be implemented to avoid cross-VM attacks due to the sharing of hardware (CPU, storage, memory, etc.). Firewalls protect the provider's internal Cloud infrastructure against insiders and outsiders, while enabling VM isolation and fine-grained filtering of addresses and ports, thus preventing DoS and DDoS attacks.

Khorshed et al. [13] organized Cloud Computing security into three sections: security categories (Cloud providers or Cloud customers), security in service delivery models : SaaS, PaaS, IaaS and security dimensions. They present a survey on the top threats for Cloud Computing and an attack detection for Cloud Computing using machine learning techniques.

Hashizume et al. [8] identify, classify, analyze and list a number of vulnerabilities, threats, mechanisms, security standards, data security, trust, security requirements for the SaaS, PaaS and IaaS delivery models of Cloud Computing. The paper enumerates the threats in detail: service hijacking, stolen data, DoS (and DDoS) and VM related issues.

Khalil et al. [14] classify Cloud security threats into five categories: Security Standards, Network, Access Control, Cloud Infrastructure and Data. They compare and analyse only countermeasures such as Intrusion Detection System (IDS) and Identity Management Systems (IMS).

Ali et al. [15] present the cloud security challenges at the communication level (between customers and cloud, communication occurring within cloud infrastructure), for Virtual machines. They discuss various approaches proposed in the literature to counter the security issues. Using tables, they indicate the security features for each countermeasure scheme.

Masdari et al. [16] presents a study of the types of DoS attacks with the new attacks against virtual machines and hypervisors in cloud computing environment. Furthermore, the authors also enumerate well known network defense and cloud computing defense against Denial of Service attacks.

Osanaiye et al. [17] survey DDoS attacks targeting cloud computing. They categorize attacks into application-bug level and infrastructural level and present the various tools to conduct these attacks.

The features of Cloud Computing (large scale, direct access to Cloud infrastructures, resource sharing, etc.) need new and innovative solutions to protect both the users and the provider. Depending on the Cloud model, security relies on the provider or on the user.

As mentioned above, Cloud Computing security is now well documented. Our survey investigates DoS and DDoS attacks specifically targeting the availability of Cloud Computing and the defense. In the following sections, for the cloud computing, our paper will present the types of attacks and attackers, the DoS and DDoS attacks, the defenses and the evaluation defense systems.

3. Attacks and Attackers

Before dealing with possible detections and mitigations of attacks on Cloud Computing, the kinds of attacks and the types of attackers that are actually a threat to Cloud Computing shall be addressed. We shall first focus on the various forms an attack can take. There are multiple scenarios involved in the Cloud infrastructure itself and its environment.

In a DDoS attack, some hosts (VM, PC or laptops), also called "bots" or "zombies", can be controlled remotely. A collection of such bots controlled by a master entity (attacker) is known as a "botnet".

The typical attackers will be classified into three categories, according to their location, their motivation or their level of activity in the attack.

3.1. Attack

Cloud Computing infrastructures can be compromised in three ways: the attack can come from the outside and the target be inside (external to internal), it can even originate from within the system (internal to internal) and it can even occur from within to target the outside of the infrastructure [18].

- *External to internal.* In such a case, the botnet used to perform the attack comes from outside the target system. The attack can target the internet gateway of the Cloud infrastructure, or the servers. If a particular client (in a VM) becomes the victim of an attack, it will also affect the other VMs present on the same physical server of the Cloud (performance interference between VMs).
- *Internal to external.* In such a case, the attack begins by taking ownership of a VM running in the Cloud. This can be done with a Trojan horse. The choice of which customer's VM to infect is important because if this customer owns a large number of VMs, the Trojan horse can potentially spread over all those VMs, therefore forming a botnet. The great computing power and resource availability of the Cloud becomes a real threat for an external target.
- *Internal to internal.* In the Cloud infrastructure, an internal botnet is formed and can attack another target inside the system (such as a VM or a group of VM). All Cloud infrastructures may break down under these kinds of attacks.

With the different kinds of attacks come different types of attackers. Indeed, each attack scenario corresponds to a particular attacker with a specific location and goals.

3.2. Attacker

The scope of an attack may greatly vary, depending on who perpetrates the attack. System administrators take the appropriate actions: to exclude or to ensure a quick recovery and allow

subsequent investigations. Raya and Hubaux [19] identify four categories of attackers that we will describe in the context of cloud computing.

- *Insider vs. Outsider*. In such a case, the insider belongs to the network that is under attack: he is an authenticated user with privileged access to critical data. Of course, the insider can do more harm than the outsider since the latter would be considered an intruder from the network perspective. Moreover, he would have fewer resources to begin an attack. In the case of Cloud Computing, an insider could be an employee of the Cloud infrastructure, or someone controlling one or several VMs inside the Cloud network, whereas an outsider would not be part of the network at all. For example, an insider attacker may be able to execute arbitrary commands on the behalf of a legitimate Cloud user, thus performing a DoS or DDoS on the user's services or to create a botnet for charging the Amazon Elastic Cloud Computing costs on the user's invoice [20].
- *Malicious or Rational*. Malicious attackers have a general goal of harming the network or the network users (employees or customers of the network). Whatever the costs or the consequences, all means can be deployed to achieve his goal and such attackers are usually harder to stop or to track since no logic is involved. On the contrary, rational attackers can be more predictable in the way the attacks are led and which specific targets are reached. Consider the example of a DoS attack in Cloud Computing: a malicious attacker may want to destabilize an organization without any claim or consistent reasons to motivate his actions: he simply wants to be famous. However, a rational attacker could be a competitor desiring to create a commercial threat or an organisation leading a DoS or DDoS against a company or a government for ideological reasons.
- *Active vs. Passive*. Active attackers lead attacks by consciously or unconsciously sending packets or signals while passive attackers may simply eavesdrop. Victims may not even be aware that their machine is under the control of a master machine that forces it to contribute to the attack (a botnet is such an example). In DoS and DDoS attacks, this defines the difference between the zombies and the master entity (active attacker): both participate in the attack, but zombies are never aware that they are vehiculing an attack. In the context of Cloud Computing, an active attacker would have taken control of one or several VMs inside the Cloud network, for instance, and would send huge amounts of traffic or malformed packets to a specific host or subnet in the network. Hence, a legitimate user such as a zombie whose VM was taken over by a master attacker, also performs the attack. A passive attacker consists on sniffing traffic to discover vulnerable links for future exploitations. In addition, passive attackers may launch eavesdropping attacks to capture the communication.
- *Local vs. Extended*. The scope of the attacker depends on the number of machines he can control. More than just a number, it really is about how those machines are linked together and scattered across the network. An attacker controlling thousands of machines outside the cloud to perpetrate a DoS or DDoS would be considered an extended attacker. On the other hand, an attacker in the Cloud, with one or several entities, would be described as local.

4. Denial of Service

A DDoS is a DoS that uses a high number of hosts to make the attack even more disruptive. The number of hosts can reach hundreds of thousands. Most of the time, the machine's owners are unaware that their machines were previously infested and corrupted through a Trojan or a backdoor program.

The actions leading to a DoS or DDoS, the ultimate goal of which is to compromise the availability of the Cloud, can take place remotely or locally from the victim's or user's service. It generally targets the victim's communication bandwidth, computational resources, memory buffers, network protocols or the victim's application processing logic.

This section specifically addresses DoS and DDoS applied to Cloud Computing networks. DoS and DDoS are not specific to Cloud networks, but they entirely apply to them.

Riquet et al. [21] study the impact of DDoS attacks on Cloud Computing with a defense such as an IDS (snort [22]) and a commercial firewall. Their experiments show that distributed attacks remain undetected, even with security solutions.

As mentioned in [23], DoS or DDoS attacks on Cloud Computing can be direct or indirect. In direct attacks, the target service or host machine is predetermined although collateral damages may result in indirect DoS or DDoSs by denying access to other services hosted on the same machine or network. There is even a scenario called race in power, induced by a Cloud mechanism that relocates flooded services to other machines. Cloud elasticity can be used to mitigate the effects of the attack, but it is entirely possible that it will simply spread the workload, in other words, direct the attack to many other servers.

Somani et al. [24] demonstrate that DDoS attacks in clouds affect the victim server along with several other parts: virtual servers on physical servers, network resources, and service providers. They conclude that these parts could be affected collaterally, even if they are not the actual targets of the attack.

According to [25], a DoS or DDoS attack can have two objectives. The first consists in overwhelming the target system resources or the network connections, by taking advantage of the superior capacity of the attacker, compared to what the system is capable of coping with in terms of CPU or bandwidth for instance. The second consists of exploiting vulnerabilities in the system by sending specific malicious packets (not necessarily at a huge rate).

4.1. Overwhelm the Resources

4.1.1. Exhausting Memory

Attacks of this category take advantage of vulnerabilities in Internet protocols, routing and networking devices. They include, for instance, SYN (SYNchronize) flood attacks that consist of sending many SYN packets, while ignoring the SYN ACK (acknowledgment) packets. Since the number of simultaneous TCP (Transmission Control Protocol) connections is limited and the server is waiting for the ACK packets, new users cannot get connected. Such attacks could be avoided with proxy-based applications for instance. The number of simultaneous TCP connections is then much higher and it decreases the server's memory load, since only the connections that have successfully completed the "three-way handshake" are forwarded to the server.

4.1.2. Exhausting Bandwidth

One way to overwhelm the target system is to exhaust the bandwidth. They aim to flood the network to prevent legitimate users from accessing the Cloud infrastructures, by imposing greater traffic than the available bandwidth. In this case, more and more packets are dropped, including the legitimate ones. An example of such an attack is given in [26]. The first step is to gain access to the topology (or at least a sufficient amount to reveal useful information such as a bottleneck uplink). According to the author, an attack has really little chance to succeed if it does not take the topology of the Cloud into account, and more specifically all of the vulnerable links. The second step is to take possession of enough hosts in the target subnet and to produce as much UDP traffic as possible through the vulnerable uplink (by targeting hosts in a specific subnet for instance). The choice of UDP is motivated by the expected starvation of the legitimate TCP sessions due to the TCP congestion handling mechanisms. In the case of CPU intensive requests, the system will predominantly process the malicious packets rather than the legitimate ones.

4.1.3. Exhausting Computing Time/Bandwidth

This attack steals computing time/bandwidth from other users. With Amazon's Cloud platform and Elastic Compute Cloud (EC2) services, an attacker boots up a massive number of machines. With a script Twill, multiple accounts are created and run the machines. This recursive registering of

accounts and booting of machines means that the number of running machines grows exponentially. This may continue until the system can no longer handle the machine load [27].

4.1.4. Exhausting Computing Time

In oversized payload attacks, an attacker sends an excessively large payload to deplete the victim's system resources. Simple Object Access Protocol (SOAP) messages from an attacker contain a large amount of references to external entities to force the server to open a large number of TCP connections to download the actual contents of the entities. Consequently, a large amount of CPU cycles is used to process the downloaded contents.

4.1.5. XML-DoS and HTTP-DoS

Those attacks belong to the resource exhaustion attack category. EXtensible Markup Language (XML) (or JSON) and HyperText Transfer Protocol (HTTP) are heavily used in Cloud Computing web services and very little work has been done to ensure security related to these protocols as, most of the time, for example with XML (XML encryption, digital signatures, user tokens, etc.), the request is implicitly assumed to be necessarily legitimate. This puts XML-DoS and HTTP-DoS among the most destructive DoS and DDoS attacks in Cloud Computing.

As Ye et al. in [28] explain, web services rely on SOAP (Simple Object Access Protocol) to send and receive messages. However, SOAP uses XML, which can be used to perpetrate XML-DoS attacks, based mainly on three strategies. The first uses an oversized payload to deplete the target system resources. The second is the External Entity DoS Attack. In this attack, the server is forced to resolve many large external entities (remote XML files) defined within the Document Type Definition (DTD). This means opening many TCP connections while making extensive use of the CPU to process the entities. Eventually, the third strategy, the XML Entity Expansion Attack, forces the server to recursively resolve entities defined within the DTD, which makes intensive usage of the CPU and the memory.

The Coercive Parsing [29] attack is one such example of XML-DoS: it uses a continuous sequence of opened tags that primarily exhausts both the CPU and the memory. Other forms of coercive parsing include many namespace declarations, a large prefix, namespace URIs, or very deeply nested XML structures [30]. However, this attack can only be successful if the web service uses a Document Object Model (DOM) parser that creates a tree representation of the XML document.

Padmanabhun et al. [31] give an overview of the underlying issues behind XML and how this can lead to a DoS. They explain how SOAP allows to send and receive XML messages regardless of the underlying implementation of the application or the transport protocol (HTTP, SMTP, etc.).

An HTTP-DoS consists of sending a lot of arbitrary HTTP requests. HTTP repeats requests and HTTP recursively attacks a particular web service [30]. A high rate of legitimate or invalid HTTP packets are sent to the server with the goal of overwhelming the web service resources. Processing all of the requests and the cost associated with each request (which may be quite significant for certain web services) eventually triggers the DoS.

4.2. Exploit the Target System's Vulnerabilities

Antunes et al. [25] give an overview of those attacks and propose a method to automatically and systematically detect the vulnerabilities that can lead to a DoS or DDoS. Those attacks are perpetrated by way of a malicious interaction with the target system. This results in either a crash or a service degradation. This can be caused by a design flaw or a software bug, for instance. As the authors point out, for system administrators, the factors leading to those kinds of attacks are very difficult to detect and therefore to be avoided, since the vulnerability may only be leveraged under very specific conditions or after many activations. They define resource-exhaustion vulnerabilities as "a specific type of fault that causes the consumption or allocation of some resource in an undefined or unnecessary way or the failure to release it when no longer needed, eventually causing its depletion".

4.3. Section Summary

Figure 1 shows a synthesis of DoS. Because of its very own nature, Cloud Computing is vulnerable to DoS and DDoS attacks, but it also offers great opportunities to recover quickly from these attacks since resources can be provisioned very easily and quickly [32]. Hence, at first sight, a DoS or DDoS attack appears to be harder to implement and its success is not granted given that the attackers need a lot more resources to achieve their goal in the case where Cloud infrastructures are well designed. However, service providers should still take those attacks into account; otherwise, Cloud elasticity would be used to serve huge amounts of illegitimate traffic, which is costly. Moreover, due to the growing botnet market (e.g., people selling access to infested machines), one cannot presume the extent of an attacker's strike force.

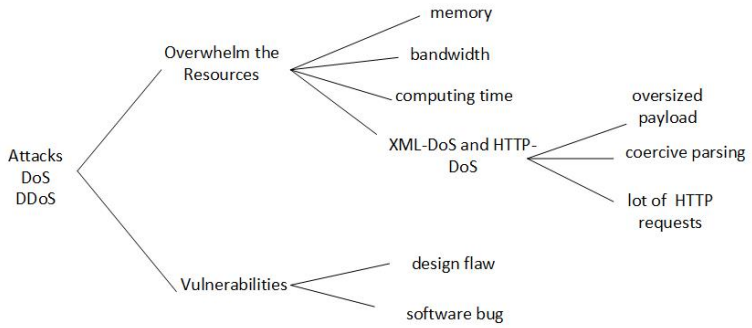

Figure 1. List of DoS (Denial-of-Service) and DDoS (Distributed Denial-of-Service) attacks.

5. DoS and DDoS Defense

The available Cloud Computing DoS and DDoS defenses cover various aspects, such as prevention, mitigation strategies and security architectures (see Table 1). During a DoS and DDoS attack, the most important thing is to maintain the availability for the service providers, the end users and the Cloud infrastructure managers.

Defending against DoS and DDoS attacks is difficult. A DoS or DDoS could theoretically be stopped by identifying and then blocking the unique source of the attack. Most of the time, the attack leverages a huge amount of bots through a DDoS attack.

Table 1. Cloud Computing DoS (Denial-of-Service) and DDoS Defenses.

		Techniques
Defense Strategy	**Prevention**	Service Level Agreement
	Attack Mitigation	Virtual Machine Monitor Intrusion Detection System IDS Firewall Detection Filter Limitation TraceBack
	Architecture	Trust Delegation Reputation Intrusion Detection System (IDS) Firewall Cryptography

5.1. Prevention

Service Level Agreements (SLA). SLA helps to prevent DoS and DDoS attacks. Kandukuri et al. [33] demonstrate the necessity of an exhaustive and standardized SLA, which is the only legal agreement between a client and the service provider for availability, confidentiality and trust. An SLA can take care of the following: (1) privileged user access, that assures the customer outsourced sensitive data do not fall into malicious hands; (2) regulatory compliance, that holds the customer ultimately responsible for his own data, and subjects the service provider to external audits and security certifications; (3) data location, that is a commitment to comply to the local jurisdiction and to store and process data in specific jurisdictions only and (4) data segregation: data must be properly encrypted to avoid leakages between users sharing the same environment. The authors also provide a list of questions that SLA must answer.

5.2. Attack Mitigation

To eradicate an attack, there are five general requirements [18]. First, detect the attack as quickly as possible and determine its magnitude (determine the impact and their level of significance). Second, try to mitigate the effects of the attack as much as possible. Third, if step two is not sufficient enough or impossible, migrate the VM under attack to safe physical servers. In order to do so, there is a fourth requirement guaranteeing network bandwidth. Eventually, put an end to the attack with countermeasures that rank from very basic to highly complex. Whatever the measure, it will not be perfect for every situation. Often a compromise must be made when choosing one or another.

As a general rule, to prevent such attacks, the resources allocated to users must be limited to a bare minimum. For authenticated users, it is possible to establish quotas to limit the load a particular user can put on the system. In particular, one might consider handling only a single request per user at one given time, by synchronizing the users' sessions. However, this solution remains problematic due to the choice of quotas and the resulting quality of service for the end user, as it may deteriorate. A more efficient solution would be to dynamically use the scalability of Cloud infrastructures to maintain availability while the attack is being eradicated.

Virtual Machine Monitor (VMM). Zhao et al. [34] propose a VMM composed of a tagger, a duplicator and a detector. The goal is to monitor and compute the amount of available resources and to compare it to a threshold to detect the presence of an attack. Since the VMM has greater privileges than the guest operating system, it can monitor and evaluate the guest's performance. When under attack, the OS and all the applications are moved to a new isolated entity. During the migration process, there is no service interruption for the user under attack since the applications are still running in both the original VM and in the new isolated VM. Basically, the only difference with duplication is that the original VM is destroyed when the migration is complete. This way, the attack has no more influence on the user's applications. The difficulty is to correctly set the threshold value that indicates an attack. Another difficulty lies in the fact that the VMM exists whether or not there is an attack. Thus, most of the time, it is likely to be idle. However, the advantage of this system is the possibility to migrate the VM without interrupting the service which is a significant advantage of this system. There is no need to migrate the entire VM, only the selected applications and OS.

Alarif et al. [35] define a strategy to detect the attack using DCT (Discrete Cosine Transform) to collect the workload signals of co-resident VMs. If the correlation between signals is higher than normal and keeps increasing with time, it strong indicates that an internal DoS attack is underway.

Intrusion Detection Systems (IDS). An IDS can be used in VMs. IDS can be classified in two categories [36]: Host-based Intrusion Detection Systems (HIDS) and Network-based Intrusion Detection Systems (NIDS). For HIDS, the detection applies for a specific host, whereas an NIDS is used for all the traffic inside a particular network. Bakshi et al. [37] propose IDSs installed on the virtual switch. With the analysis of inbound-outbound traffic, the IDS blocks the intruder's addresses. They suppose that the IP addresses are not spoofing.

IDS, Behavior and Knowledge Analysis. Vieira et al. [38] propose an architecture (node, service, event auditor, storage service) for IDS to examine network traffic, log files and user behaviours. Each node must alert other nodes when an attack occurs. A node contains the resources (through middleware), the service provides functionality, the event auditor monitors the data to analyze and the storage service uses behavior and knowledge analysis: data mining, artificial neural networks, artificial immunological systems and expert systems. Data from both the logs and the communication systems are used to evaluate the Knowledge-Based System. A series of rules was created to build a security policy that should be respected.

IDS and Cloud Fusion Unit. Lonea et al. [39] propose a solution to combine IDSs deployed in VMs of the Cloud system with a data fusion methodology at the front-end using the Dempster's combination rule (Dempster–Shafer Theory DST). The IDSs are installed and configured in each VM. A MySQL database is installed in the Cloud Fusion Unit (CFU) of the front-end server. An alert in IDSs will be stored in the database. The Cloud Fusion Unit (CFU) comprises three components: a MySQL database (storing the alerts), basic probabilities assignment calculation operations and attacks assessment. Their solution is not associated with any experimentation.

IDS and Queueing Theory. Yu et al. [40] propose an Intrusion Prevention System (IPS) between a Cloud data center and the internet to monitor incoming packets. To mitigate DoS or DDoS attacks on individual Cloud customers, the mechanism will automatically and dynamically allocate extra resources from the available Cloud resources pool. A queuing theory is used to estimate the resource allocation. The mitigation problem is an optimization problem: minimizing the resource investment (CPU, memory, IO, bandwidth) while guaranteeing the average time in the system of packets. However, some statements in the paper are false: (1) the attack capability of a botnet is usually limited. Consequently, the authors find it reasonable to expect that a Cloud can manage its reserved or idle resources to meet demand; (2) all attack packets are filtered and all legitimate packets go through the IPS system.

Firewalls. Modi et al. [36] explain that firewalls protect the front access points of Clouds and are treated as the first line of defense. Firewalls filter (1) by inspecting only header information such as source or destination address and the port number; (2) with a state table (request and server responses); and (3) by analyzing the protocol syntax by breaking off the client/server connection.

Ismail et al. [41] propose a framework that web servers in virtual machine access by internet gateway and virtual switch. With a covariance matrix of normal traffic, the virtual switch can find the IP addresses from where the attacks originated. Then, the virtual switch blocks the IP addresses that perpetrated the attacks with a honeypot network. The authors suppose that the IP addresses at the origin of the attacks are not spoofing.

Clusterized firewall. Liu et al. [42] propose a clusterized firewall framework for Cloud Computing. They divide the Cloud services into application layers in which the servers are grouped into clusters, for a type of Cloud data service center. Each cluster has a firewall. The firewall for each cluster protects applications according to the arrival rate and thus guarantees QoS for legitimate users. Each cluster can be modeled as an M/G/1 queueing system to obtain the key measures: (1) the request response time and (2) how many resources are needed to guarantee the QoS. These key metrics evaluated the Cloud defense.

Statistical machine learning. With statistical machine learning techniques, Khoshed et al. [13] propose a Support Vector Machine technique to identify top attacks. It should also warn the system administrators and data owners of the type of attack and suggest possible actions to take. Eventually, customers would be aware of the attack type even if Cloud providers are reluctant to divulge information about the attack.

Traceback and Cloud filter. Yang et al. [43] propose an SOA-based tracing approach to trace the true DDoS source and filter it. The SOA is placed before the Web server and all requests for the service are marked by the SOA-Based Traceback Approach (SBTA). An algorithm for determining the

reconstruction of the path makes it possible to find the true source of attack. The attack messages are then filtered by the Cloud.

Infrastructure. Amazon Web Services (AWS) [44] proposes an Infrastructure Layer Defense and an Application Layer Defense against DDoS. The Infrastructure Layer defense (1) uses a resizable compute capacity; (2) chooses AWS Regions for optimal latency and throughput; (3) considers Elastic Load Balancing (ELB); (4) uses a content delivery network (CDN) service and (5) uses a scalable domain name system (DNS) service. The Application Layer defense uses Web application firewalls to protect the vulnerabilities within the application, or to block the unknown source IP addresses, URI, query string, HTTP method, or header key.

5.3. Security Architecture

A security architecture involves several elements: servers, switch controller, protocols, router and applications.

Security Aware Cloud Architecture. Hwang et al. [45] highlight that the abstraction level of the Cloud model (SaaS being the most abstract and IaaS being the least abstract) influences the number of security aspects that will be handled by the provider (more abstraction means the provider will be in charge of more security aspects). In the intermediate case of PaaS, users remain in charge of confidentiality and data privacy, yet the provider is responsible for data integrity and availability. Generally speaking, they propose a Security Aware Cloud Architecture that offers protection to secure public Clouds and data centers. The mechanisms are (1) trust delegation and negotiation architecture; (2) worm containment and DDoS defense; (3) reputation system of resource sites; (4) fine-grain access control and (5) collusive privacy prevention. No tests assess the performance of their architecture.

DDoS Attack Mitigation Architecture: DaMask. Wang et al. [46] propose a DDoS attack mitigation architecture. The Software-Defined Networking (SDN) is an approach that allows network administrators to manage network services by way of abstraction of lower-level functionality. The authors find that the SDN and Cloud Computing can enhance the DDoS attack defense. The DaMask architecture has three layers: network switches, network controllers and network applications. There are two separate modules: (1) DaMask-D, a network attack detection system, and (2) DaMask-M, an attack reaction module. DaMask-D already has an efficient attack detection algorithm with a very low overhead. DaMask-M defines three basic operations: forward, drop and modify the packet. Those operations are implemented as a set of APIs. Consequently, the defenders can customize the countermeasures.

Security Architecture based on Defense. Mavroeidakos et al. [47] propose the security architecture composed of different types of firewalls that cooperate in defense zones. It includes a set of layers: (1) the perimeter defense; (2) the deceptive; (3) the detection and (4) the cryptography. The security mechanisms proposed in each layer are implemented in the defense zones and collaborate to protect the data. The perimeter defense consists of a border router and two stateful firewalls. Between the provided service and the rest of the Internet, this defense provides the core security functionality to protect the classified data in defense zones. The deceptive layer operate in every defense zone with a honeynet to identify new attacks and vulnerabilities of their systems. The detection layer consists of IDSs that analyze the network traffic with a predefined rule set. The cryptography layer into the cloud environment is the elliptic curve cryptography. This attacks the following: address spoofing, tiny fragment attacks, buffer overflows, port scans, OS fingerprinting, web attacks, Trojan attacks, viruses and worms, insider threats, attacks on virtualization, DoS and DDoS attacks, which are mitigated by a number of security components in the architecture. The authors do not evaluate the cost of implementing such an architecture.

Latanicki et al. [18] propose a federated cloud architecture to use migration of virtual machines to defense against cloud DDoS attacks. The architecture is composed of (1) Scalable Cloud DDoS Probe Manager that is responsible for monitoring the user Internet access; (2) Scalable Cloud DDoS Correlation Analysis Manager is responsible for performing correlation analysis in each Cloud

infrastructure; (3) DDoS Cloud Migration Manager is responsible for the availability by migrating the attacked Virtual Machine to physical machines not under DDoS attack; and (4) DDoS Cloud ReRoute Manager is responsible for the network interconnections.

Hybrid Firewalling Architecture. Guenane et al. [48] present a DDoS mitigation service using an innovative architecture that provides hybrid (Physical and Virtual) cloud-based firewalling services. The physical firewalls represent the physical IT-security infrastructure of the company. The mitigation service aims at redirecting or load balancing the traffic which is redirected to virtual firewalls. The virtual firewalls reside on virtual machines and execute several operations such as analysis, monitoring, and reporting.

(1) The Communication Module performs the authentication and establishment of a secure tunnel between the physical and virtual firewalls.
(2) The Decision Strategy Module determines if traffic must be transferred to virtual firewalls in order to decrease the overload on physical ones. This decision depends on the information provided by the Monitoring System and Monitoring Network modules such as CPU, memory and throughput.
(3) The Load Balancing Module receives its orders from the Decision Strategy Module. It interacts with the Authentication Module that provides it with trusted information (IP-address and port number).
(4) The Monitoring Module is constantly polling the virtual firewall to get an accurate overview of software and network status.
(5) The Evaluation Module aims at evaluating the status of the virtual firewall that is provided by the Monitoring Module by aggregating different related parameters.

5.4. Defenses against XML-DoS and HTTP-DoS Attacks

Filtering Tree. Karnwal et al. [49] developed a filtering tree, which works like a service. The XML consumer request is converted into a tree form and uses a virtual Cloud defender to defend against these types of attacks. The Cloud defender basically consists of five steps: sensor filtering (check number of messages from a particular user), hop count filtering (number of nodes crossed from source to destination—this cannot be forged by the attacker), IP frequency divergence (the same range of IP addresses is suspect), puzzle (it sends a puzzle to a particular user: if it is not resolved, the packet is suspect) and double signature. The first four filters detect HTTP-DDoS attacks while the fifth filter detects XML-DDoS attacks.

Limitation. Karthigeyan et al. [50] explain that an acceptable solution to prevent attackers from exhausting the victims' network bandwidth and computing power is to route the requests to the service providers only once they have been authenticated and validated. First, limit the payload size. Then, limit the time allocated to a SOAP request. Third, limit the number of requests a particular user can send within a given time frame. Packets that do not match those criteria are discarded and the service is blocked for the user for a certain period of time. They also propose to impose limits for the XML parser. For example, limit the number of attributes an element can have, the quantity of bytes in a XML message, the depth of nested elements and the size of all nodes in the XML document. Furthermore, to minimize the impact on the QoS for the end user in terms of delays, for instance, this could take place only when the system is under attack, which is detected by the service provider.

Cloud Protector and Decision Theory. Chonka et al. [51,52] developed a Cloud Traceback (CTB), which uses a Service-Oriented Traceback Architecture (SOTA) approach. CTB is deployed at the edge routers in order to be close to the Cloud network source end to mark all outgoing packets. If an attack is detected, the use of a back propagation neutral network, called Cloud Protector, allows for retrieving the source of the attack. The Cloud Protector is a trained back propagation neural network, which means that there is a set of connected units associated with a given weight, spread between input, hidden and output layers. Then, the weights are added to see if the result exceeds a certain threshold, which means an attack is taking place. They also developed a method relying on decision

theory, called ENDER (Pre-Decision, Advance Decision and Learning System). It uses two decision theory methods to detect attack traffic and mark the attack messages. If an attack message is detected, a Reconstruct And Drop (RAD) system removes the message before a victim is harmed.

Defense in Cloud Broker. Vissers et al. [30] divide the concept of Cloud Computing into three parts. First, the Cloud providers deploy the VM and their web services. Then, the Cloud broker makes the link between the user and the available resources of the Cloud providers in order to allocate the necessary resources. Users request resources in the Cloud infrastructure through the Cloud broker to eventually use the web services hosted by the Cloud providers. As the author points out, the Cloud broker introduces a single point of failure, since its unavailability makes the Cloud infrastructure unusable. To make this architecture more secure, a DDoS defense system is placed with the Cloud broker to decide whether the application request should be rejected or not. The defense system, which uses DDoS datasets, is incorporated into all the broker entities. The filter is based on the definition of a normal profile usage constructed with previous requests. The filter, aimed to be scalable to overcome a DDoS, is transparent for the user. A request must go through the HTTP header filtering (HTTP floods, non-existing SOAP Action usage and content-length outliers) and then through actual XML content filtering (SOAP feature outlier detection and SOAP Action or WS-Addressing spoofing). This defense mechanism has proved to be successful at detecting and mitigating all listed vulnerabilities. In addition, it might even be able to handle unknown vulnerabilities with minimal time overhead.

Flexible, Collaborative, Multilayer, DDoS Prevention Framework (FCMDPF). Saleh et al. [53] propose a framework composed of (1) an Outer attack Blocking (OB) at the edge router; (2) a Service Traceback Oriented Architecture (STBOA); (3) and a flexible advanced entropy based (FAEB) layer. From a blacklist database table (IP source), the OB layer blocks or forwards the incoming request. The STBOA layer is designed to validate whether the incoming request is launched by a human (real web browser) or by an automated tool (bots). A puzzle or random number is sent to the client or the requester to solve. After verification, if the puzzle or random number are correct, the request is forwarded to the next level. Otherwise, it is immediately blocked and a blacklist is updated. The FAEB layer computes entropy of overall requests to determine flash crowds or HTTP attacks. The entropy of incoming requests that are launched towards hot pages of the website determines the flash crowd. In the case of an HTTP attack, the blacklist is updated. The disadvantage of this framework lies in the information on the blacklist and its updates.

Using CAPTCHA to Mitigate HTTP-DoS Attacks. Sairam et al. [54] propose an architecture to migrate HTTP-DoS attacks with (1) a feature extractor; (2) a clustering unit; (3) a workload calculator and (4) a filter. With a navigation log, the extractor module extracts: browsing length ratio, average stay time and the HTTP request rate to identify user groups. Every T interval, the clustering module samples the user traffic. Experiments show that for T = 15 min offers a good performance. By using the clustering report, the workload calculator computes the size and density of each cluster. The load is calculated both with the computational overhead (pageload factor) and the network bandwidth usage (number of bytes transferred for each request). Finally, if the workload exceeds a certain threshold, for each cluster, the filter provides a mechanism to filter traffic from such sources. The difficulty is to set the value of threshold as it should be proportional to the capacity of the web server. If the workload exceeds a threshold and a cluster is highly dense, then the users are suspected to be bots. The potential attackers are then challenged with a CAPTCHA to eliminate false positives.

5.5. Section Summary

The defenses to protect the Cloud availability need some information: blacklist, the normal profile, threshold determination and limit fixation. Hence, if those information values are wrong, the defenses can lead to false positives or negatives.

6. How to Evaluate Defense Systems

The proposed defenses for DDoS attacks must prove their efficiency. For this purpose, research authors must design experiments and determine metrics to assess performance. Table 2 lists all the information for the proposed defenses. Some defenses are compared with others. However, the following authors have tried to implement, simulate and test their proposed solutions.

Table 2. Summary of Evaluation Defense System in DDoS Attacks.

Experiment	Efficiency
Design	**Impact of defense system**
Data collection	Overhead bandwith
Simulation	Processing time
Testbed	Detection attack
	False alarm
	False negative
Parameters	**Mitigation**
Attack Rate	Filtration
Attack Duration	Limitation
	Quality of Serviice (QoS)
	Migration Virtual Machine (VM)
	Traceback

6.1. Theoretical Evaluation

Zhao et al. [34] propose to monitor the VM's available resources to decide if it is under attack, in which case the system selectively duplicates tagged applications and operating systems. Their defense system has yet to be tested, but the authors have identified four benefits and one disadvantage with this method. The isolated environment exists whether or not there is an attack, so that most of the time it is likely to be idle. From a performance point of view, they show that their system is not necessarily more costly than VM migration. As far as performance is concerned, they theoretically evaluate the total time consumption of their defense system.

6.2. Data Collection

Vieira et al. [38] propose a refinement to traditional IDS to be more efficient in a Cloud environment. To test their system, they use three sets of data. The first represents legitimate actions. In the second, they altered the services and their usage frequency to simulate anomalies. The last set simulates policy violations. In order to evaluate the event auditor that monitors the requests received and the responses sent on a node, they chose to examine the communication elements, since log data present little variations, making attacks difficult to detect. A feed-forward neural network is used for the behavior-based technique, and the simulation includes five legitimate users and five intruders. Their scenario simulates ten days of usage. Although the results yielded a high number of false negatives and positives, its performance improved when the training period of the neural network was prolonged. To evaluate their behavior-based system, they looked for the number of false positives and negatives. They show that their system consistently has more false negatives than false positives. With still a high level of uncertainty, the false alarms disappeared within 16 days of simulation training. With longer training periods, they noticed even lower false positives, but also the very non-deterministic nature of neural networks, since false positives were not stable after several iterations. To evaluate the Knowledge-Based System, they used data from both the logs and the communication system. They created a series of rules to build a security policy that should be respected. They conclude that their system could allow real time analyses, provided the number of rules per action remains low.

6.3. Simulation

Liu et al. [42] use three parameters: the attack rate, the attack duration and the rule processing time. They showed that a larger matching probability (that is to say rules that are easier to match) means a reduced response time. Hence, they encouraged Cloud defenders to put those rules at the top of the rules list so as to increase users' satisfaction. They demonstrated, both analytically and experimentally, a direct correlation between the response time and the number of rules and attack rates. To estimate the cost of their system, they rented 20 VMs from Amazon EC2. In the end, running their clusterized firewall turned out to cost 38 US/day and 266 US/week, while keeping in mind that long attacks are extremely rare, given that they are easily detected.

6.4. Testbed

Wang et al. [46] set up a hybrid Cloud, using Amazon EC2 as a public Cloud and simulating a private Cloud in their lab. The private Cloud consists of two Linux machines, one of which hosting DaMask and the network controller while the other emulates a virtual network to extend the private Cloud. They wanted to measure the communication costs as well as the computation overhead of DaMask. They began by computing the network bandwidth between the private and the public Cloud, with and without DaMask. For the communication overhead, caused by the traffic being examined by DaMask, they showed the overhead was a constant if the link status of network remained stable. As for the computation overhead caused by the detection algorithm, they ended up with an interference time of 80 ms, which they consider quite efficient.

Chonka et al. [51] evaluated their Cloud Traceback (CTB) and Cloud Protector in multiple ways. They conducted experiments to see how CTB marked packets and could determine if they led to a XML-DoS, but also how accurate CTB was at identifying the source of the attack. Moreover, they wanted to make sure their method was better than traditional security mechanisms such as WS-Security, when it comes to XML-DoS attacks. The experimentation for the Cloud Protector consisted in determining if it could detect and filter HTTP-DoS and XML-DoS messages. To generate the attack traffic, they opened up three virtual servers that contained 20 Firefox browsers and 20 open tabs to each browser, in addition to a page refresher tool and targeted a particular website. First, they studied the impact of the attack without their architecture and witnessed its tremendous and lasting effect as the web server was quickly unable to handle more than a few requests. Then, they divided the evaluation of CTB in two parts: one to simulate XML-DoS attacks against a web server and the other to compare CTB to traditional mechanisms. In their simulation, 100 messages were sent. If one was an XML-DoS attack, it was supposed to crash the server with a probability of 1/2. Out of those 100 messages, nine successfully crashed the server and CTB was able to identify seven of them, with a response time that varied between 480 and 550 ms. When comparing CTB with SOAP authentication, CTB proved to be far more effective in terms of response time and the same was true for WS-Security. Then, they assessed the performance of the Cloud Protector (a neural network) by training and testing it with a dataset they developed. On the trained dataset for HTTP-DoS, the Cloud Protector was able to identify 91% of the attack (9% of false positives), but with significantly different response times (between 20 ms and 1 s). On the test dataset, the accuracy decreased to 88%, with the same variation in response time. For XML-DoS, most of the attack messages were identified, detecting and removing them taking between 10 and 140 ms, but the response times were not as scattered as in the HTTP-DoS case.

In [52], Chonka et al. came up with a new defense system called ENDER, that has no more than 1% of false positives on the same dataset.

Vissers et al. [30] used the Eucalyptus middleware to manage resources in their experiment. The Cloud Resource Broker along with the DDoS defense system is installed on a four CPU server. Then, they set up Eucalyptus and the web services. They wanted to assess the impact of the attack, the mitigation capabilities of their system and the cost induced by this defense mechanism (extra response time). When the defense system was not present, a flooding attack with 4900 legitimate

requests during 60 s was enough to exhaust the CPU, whereas with the defense system, only the first request was accepted and the others were discarded as they exceeded the request rate. Moreover, some addresses were blacklisted. All in all, 10% of CPU was necessary to drop malicious packets. With an oversized XML attack, ten 12 MB messages were sufficient to keep the CPU busy for 15 min, while also significantly increasing memory usage. With the defense system, the CPU load was only around 10% for 20 s, while blocking the attack, and without additional memory usage. The mechanism also successfully handled a coercive parsing attack, an oversize encryption attack and a spoofing attack. To evaluate the response time of their system, they sent 200 SOAP requests at a rate of two requests per second and then 10,000 SOAP requests at 50 requests per second, both with and without the filter in place. In the first load, introducing the defense system made the processing time of the web service go from 3 to 5 ms, whereas in the high load, it went from 3 to 9 ms. Those results outperform the current existing solutions (for example, the Cloud Protector mentioned above).

Saleh et al. [53] conducted four experiments. First, they evaluated how the framework could protect against flash crowd attacks, then against high rate DDoS attacks. Third, they studied the ability to validate clients and trace the true IP source of the attack. Eventually, they evaluated the proper blocking of the attacking IP address as close as possible to the network entrance. They used the following parameters to make the evaluations: the number of incoming requests and number of detected and prevented attacks against the web application (conducted by way of the Apache log, based on the response code number to the incoming request—an error code means the attack was detected). Finally, 420,000 incoming requests were generated to cause a DDoS. The AntiDDoS_Shield system detected and prevented all high rate HTTP-based DoS/DDoS attacks: 369,726 out of 420,000 flash crowd (FC) attacks, at the edge router. AntiDDoS_Shield system succeeded in validating and tracing back 369,726 out of 420,000 incoming requests.

In summary, apart from defense architectures that have no evaluations whatsoever, most defenses came with experiments on design and performance, whether theoretical or practical. By using several sets of data and carefully choosing the parameters of interest, the authors were able to study the response of their system under different workloads, while reproducing real-life situations. They discussed the possibility of adopting their system regarding the cost, scenario (real time), best practices (how to correctly set up the system for maximal performance), overhead, precision (rate of false positives or negatives), effectiveness to detect and filter attack messages and ability to detect new attacks. Halabi et al. [55] propose a survey and a taxonomy to evaluate the defense against DoS and DDoS attacks in Cloud Computing.

7. Conclusions

Being a combination of existing technologies such as VM, web services, servers, network links, etc., this new paradigm Cloud Computing comes with known vulnerabilities, but also new kinds of attacks because of the innovative way services are presented to the user and because of the growing success and adoption of Cloud Computing, both by companies and individuals. Taking advantage of its great scalability and elasticity, Cloud Computing apparently offers adequate resistance to attacks. This review proves that many attacks can still cause great harm to Cloud Computing, impacting all the important security aspects (confidentiality, integrity, isolation, availability, etc.). Among those attacks, the DoS and DDoS attacks are arguably the easiest to mount and the most destructive, yet huge gaps still exist to efficiently deal with those attacks. We presented some state-of-the-art solutions: some were rather easy to incorporate in existing Cloud infrastructures for Cloud providers to prevent or reduce DoS and DDoS attacks. However, some solutions could not detect nor perfectly mitigate all the possible attacks. Others were much more efficient, albeit much more complex. In all cases, and as always in the security field, no solution is perfect. Eventually, it all comes down to what compromise the system administrators are willing to make. We also gave an original focus on the different facets of the attack and attacker applied to Cloud Computing, a key parameter to know in order to provide the best security solutions. The extensive study of XML-DoS and HTTP-DoS allowed for showing all of the

available countermeasures. In addition, DoS and DDoS defenses are evaluated with the appropriate metrics and experimental design.

For the cloud provider, this survey of DoS and DDoS attack and defenses can help to define a Security Service Level Agreements (Security-SLAs), as the protection of the fundamental security attributes is defined by the CIA triad (Confidentiality, Integrity and Availability).

Acknowledgments: Funds for covering the costs to publish in open access are from the Department of Computer Engineering and Software Engineering of École Polytechnique of Montréal.

Author Contributions: All authors Adrien Bonguet and Martine Bellaiche have read, analyzed the articles and synthesized.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zissis, D.; Lekkas, D. Addressing cloud computing security issues. *Future Gener. Comput. Syst.* **2012**, *28*, 583–592.
2. Sridhar, T. Cloud Computing: Infrastructure and Implementation Topics. *Int. Protoc. J. CISCO* **2009**, *12*, 4.
3. Los, R.; Gray, D.; Shackleford, D.; Sullivan, B. *The Notorious Nine: Cloud Computing Top Threats in 2013*; CSA, Cloud Security Alliance: 2013. Available online: https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf (accessed on 4 August 2017).
4. Ristenpart, T.; Tromer, E.; Shacham, H.; Savage, S. Hey, You, Get off of My Cloud: Exploring Information Leakage in Third-party Compute Clouds. In *CCS'09, Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, USA, 9–13 November 2009*; ACM: New York, NY, USA, 2009; pp. 199–212.
5. Amazon, EC2. Available online: https://aws.amazon.com/ec2 (accessed on 4 August 2017).
6. Service, A.W. 2017. Available online: https://aws.amazon.com/compliance/data-privacy-faq/ (accessed on 4 August 2017).
7. Hewlett-Packard. Security Overview of the Integrity Virtual Machines Architecture. Available online: http://h20564.www2.hpe.com/hpsc/doc/public/display?docId=emr_na-c02018861&DocLang=en&docLocale=en_US (accessed on 4 August 2017).
8. Hashizume, K.; Rosado, D.; Fernandez-Medina, E.; Fernandez, E. An analysis of security issues for cloud computing. *J. Int. Serv. Appl.* **2013**, *4*, 5.
9. Gonzalez, N.; Miers, C.; Redigolo, F.; Carvalho, T.; Simplicio, M.; de Sousa, G.; Pourzandi, M. A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing. In Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), Athens, Greece, 29 November–1 December 2011; pp. 231–238.
10. KrebsonSecurity. DDoS on Dyn Impacts Twitter, Spotify, Reddit. Available online: https://krebsonsecurity.com/2016/10/ddos-on-dyn-impacts-twitter-spotify-reddit/ (accessed on 3 August 2017).
11. Khandelwal, Massive DDoS Attacks against Dyn DNS 2016. Available online: http://thehackernews.com/2016/10/dyn-dns-ddos.html (accessed on 3 August 2017).
12. Grobauer, B.; Walloschek, T.; Stocker, E. Understanding Cloud Computing Vulnerabilities. *Secur. Priv. IEEE* **2011**, *9*, 50–57.
13. Khorshed, M.T.; Ali, A.S.; Wasimi, S.A. A Survey on Gaps, Threat Remediation Challenges and Some Thoughts for Proactive Attack Detection in Cloud Computing. *Future Gener. Comput. Syst.* **2012**, *28*, 833–851.
14. Khalil, I.M.; Khreishah, A.; Azeem, M. Cloud computing security: A survey. *Computers* **2014**, *3*, 1–35.
15. Ali, M.; Khan, S.U.; Vasilakos, A.V. Security in cloud computing: Opportunities and challenges. *Inf. Sci.* **2015**, *305*, 357–383.
16. Masdari, M.; Jalali, M. A survey and taxonomy of DoS attacks in cloud computing. *Secur. Commun. Netw.* **2016**, *9*, 3724–3751; SCN-15-0746.R1.
17. Osanaiye, O.; Choo, K.K.R.; Dlodlo, M. Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. *J. Netw. Comput. Appl.* **2016**, *67*, 147–165.
18. Latanicki, J.; Massonet, P.; Naqvi, S.; Rochwerger, B.; Villari, M. Scalable Cloud Defenses for Detection, Analysis and Mitigation of DDoS Attacks. In *Towards the Future Internet*; IOS Press: Amsterdam, The Netherlands, 2010; pp. 127–137.

19. Raya, M.; Jean-Pierre, H. Securing vehicular ad hoc networks. *J. Comput. Secur.* **2007**, *15*, 39–68.
20. Gruschka, N.; Iacono, L. Vulnerable Cloud: SOAP Message Security Validation Revisited. In Proceedings of the IEEE International Conference on Web Services, Los Angeles, CA, USA, 6–10 July 2009; pp. 625–631.
21. Riquet, D.; Grimaud, G.; Hauspie, M. Large-Scale Coordinated attacks: Impact on the Cloud Security. In Proceedings of the 2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Palermo, Italy, 4–6 July 2012; pp. 558–563.
22. Roesch, M. Snort-Lightweight Intrusion Detection for Networks. In Proceedings of the LISA'99 13th USENIX Conference on System Administration, Berkeley, CA, USA, 7–12 November 1999; USENIX Association: Berkeley, CA, USA, 1999; pp. 229–238.
23. Fernandes, D.A.B.; Soares, L.F.B.; Gomes, J.V.P.; Freire, M.M.; Inácio, P.R.M. Security issues in cloud environments: A survey. *Int. J. Inf. Secur.* **2014**, *13*, 113–170.
24. Somani, G.; Gaur, M.S.; Sanghi, D.; Conti, M. {DDoS} attacks in cloud computing: Collateral damage to non-targets. *Comput. Netw.* **2016**, *109 Pt 2*, 157–171. Traffic and Performance in the Big Data Era.
25. Antunes, J.; Neves, N.; Verissimo, P. Detection and Prediction of Resource-Exhaustion Vulnerabilities. In Proceedings of the 19th International Symposium on Software Reliability Engineering, Seattle, WA, USA, 10–14 November 2008; pp. 87–96.
26. Liu, H. A New Form of DOS Attack in a Cloud and Its Avoidance Mechanism. In CCSW'10, Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop, Chicago, IL, USA, 8 October 2010; ACM: New York, NY, USA, 2010; pp. 65–76.
27. Presentation Demo vids: Amazon, B. 2009. Available online: https://sensepost.com/blog/2009/blackhat-presentation-demo-vids-amazon/ (accessed on 4 August 2017).
28. Ye, X. Countering DDoS and XDoS Attacks against Web Services. In Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Shanghai, China, 17–20 December 2008; Volume 1, pp. 346–352.
29. WS-Attacks.org. Coercive Parsing. Available online: http://www.ws-attacks.org/Coercive_Parsing. (accessed on 3 August 2017).
30. Vissers, T.; Somasundaram, T.S.; Pieters, L.; Govindarajan, K.; Hellinckx, P. DDoS defense system for web services in a cloud environment. *Future Gener. Comput. Syst.* **2014**, *37*, 37–45.
31. Padmanabhuni, S.; Singh, V.; Senthil Kumar, K.; Chatterjee, A. Preventing Service Oriented Denial of Service (PreSODoS): A Proposed Approach. In Proceedings of the 2006 International Conference on Web Services, Chicago, IL, USA, 18–22 September 2006; pp. 577–584.
32. Fox, A.; Griffith, R.; Joseph, A.; Katz, R.; Konwinski, A.; Lee, G.; Patterson, D.; Rabkin, A.; Stoica, I. *Above the Clouds: A Berkeley View of Cloud Computing*; Technical Report No. UCB/EECS-2009-28; Department Electrical Engineering and Computer Sciences, University of California at Berkeley: Berkeley, CA, USA, 2009; Volume 28, p. 13.
33. Kandukuri, B.; Paturi, V.; Rakshit, A. Cloud Security Issues. In Proceedings of the 2009 IEEE International Conference on Services Computing, Bangalore, India, 21–25 September 2009; pp. 517–520.
34. Zhao, S.; Chen, K.; Zheng, W. Defend Against Denial of Service Attack with VMM. In Proceedings of the 2009 Eighth International Conference on Grid and Cooperative Computing, Lanzhou, China, 27–29 August 2009; pp. 91–96.
35. Alarifi, S.; Wolthusen, S.D. Mitigation of Cloud-Internal Denial of Service Attacks. In Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering, Oxford, UK, 7–11 April 2014; pp. 478–483.
36. Modi, C.; Patel, D.; Borisaniya, B.; Patel, H.; Patel, A.; Rajarajan, M. A survey of intrusion detection techniques in Cloud. *J. Netw. Comput. Appl.* **2013**, *36*, 42–57.
37. Bakshi, A.; Dujodwala, Y.B. Securing Cloud from DDOS Attacks Using Intrusion Detection System in Virtual Machine. In Proceedings of the 2010 Second International Conference on Communication Software and Networks, Singapore, 26–28 February 2010; pp. 260–264.
38. Vieira, K.; Schulter, A.; Westphall, C.; Westphall, C. Intrusion Detection for Grid and Cloud Computing. *IT Prof.* **2010**, *12*, 38–43.
39. Lonea, A.M.; Popescu, D.E.; Tianfield, H. Detecting DDoS Attacks in Cloud Computing Environment. *Int. J. Comput. Commun.* **2013**, *8*, 70–78.

40. Yu, S.; Tian, Y.; Guo, S.; Wu, D. Can We Beat DDoS Attacks in Clouds? *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2245–2254.
41. Ismail, M.N.; Aborujilah, A.; Musa, S.; Shahzad, A. New Framework to Detect and Prevent Denial of Service Attack in Cloud Computing Environment. *Int. J. Comput. Sci. Secur.* **2012**, *6*, 226–237.
42. Liu, M.; Dou, W.; Yu, S.; Zhang, Z. A clusterized firewall framework for cloud computing. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, Australia, 10–14 June 2014; pp. 3788–3793.
43. Yang, L.; Zhang, T.; Song, J.; Wang, J.S.; Chen, P. Defense of DDoS attack for cloud computing. In Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE), Zhangjiajie, China, 25–27 May 2012; Volume 2, pp. 626–629.
44. Services, A.W. AWS Best Practices for DDoS Resiliency. 2016. Available online: https://d0.awsstatic.com/whitepapers/DDoS_White_Paper_June2015.pdf (accessed on 3 August 2017).
45. Hwang, K.; Kulkareni, S.; Hu, Y. Cloud Security with Virtualized Defense and Reputation-Based Trust Mangement. In Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing, Chengdu, China, 12–14 December 2009; pp. 717–722.
46. Wang, B.; Zheng, Y.; Lou, W.; Hou, Y. DDoS Attack Protection in the Era of Cloud Computing and Software-Defined Networking. In Proceedings of the 2014 IEEE 22nd International Conference on Network Protocols (ICNP), Raleigh, NC, USA, 21–24 October 2014; pp. 624–629.
47. Mavroeidakos, T.; Michalas, A.; Vergados, D.D. Security architecture based on defense in depth for Cloud Computing environment. In Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, USA, 10–14 April 2016; pp. 334–339.
48. Guenane, F.; Nogueira, M.; Pujolle, G. Reducing DDoS attacks impact using a hybrid cloud-based firewalling architecture. In Proceedings of the 2014 Global Information Infrastructure and Networking Symposium (GIIS), Montreal, QC, Canada, 15–19 September 2014; pp. 1–6.
49. Karnwal, T.; Sivakumar, T.; Aghila, G. A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack. In Proceedings of the 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 1–2 March 2012; pp. 1–5.
50. Karthigeyan, A.; Andavar, C.; Jaya Ramya, A. Adaptable Practices for Curbing XDoS Attacks. *Int. J. Sci. Eng. Res.* **2012**, *3*, 1073–1078
51. Chonka, A.; Xiang, Y.; Zhou, W.; Bonti, A. Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. *J. Netw. Comput. Appl. Elsevier* **2011**, *34*, 1097–1107.
52. Chonka, A.; Abawajy, J. Detecting and Mitigating HX-DoS Attacks against Cloud Web Services. In Proceedings of the 2012 15th International Conference on Network-Based Information Systems (NBiS), Melbourne, Australia, 26–28 September 2012; pp. 429–434.
53. Saleh, M.A.; Manaf, A.A. A Novel Protective Framework for Defeating HTTP-Based Denial of Service and Distributed Denial of Service Attacks. *Sci. World J.* **2015**, *2015*, 238230.
54. Sairam, A.S.; Roy, S.; Dwivedi, S.K. Using CAPTCHA Selectively to Mitigate HTTP-Based Attacks. In Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, USA, 6–10 December 2015; pp. 1–6.
55. Halabi, T.; Bellaiche, M. How to Evaluate The Defense Against DoS and DDoS Attacks in Cloud Computing: A Survey and Taxonomy. *Int. J. Comput. Sci. Inf. Secur.* **2016**, *14*, 1–10.